Hummer *et al. EURASIP Journal on Information Security* (2016) 2016:19
DOI 10.1186/s13635-016-0043-2

EURASIP Journal on Information Security

RESEARCH Open Access

Adaptive identity and access management—contextual data based policies

Matthias Hummer[1,2*], Michael Kunz[2], Michael Netter[1], Ludwig Fuchs[1] and Günther Pernul[2]

Abstract

Due to compliance and IT security requirements, company-wide identity and access management within organizations has gained significant importance in research and practice over the last years. Companies aim at standardizing user management policies in order to reduce administrative overhead and strengthen IT security. These policies provide the foundation for every identity and access management system no matter if poured into IT systems or only located within responsible identity and access management (IAM) engineers' mind. Despite its relevance, hardly any supportive means for the automated detection and refinement as well as management of policies are available. As a result, policies outdate over time, leading to security vulnerabilities and inefficiencies. Existing research mainly focuses on policy detection and enforcement without providing the required guidance for policy management nor necessary instruments to enable policy adaptibility for today's dynamic IAM. This paper closes the existing gap by proposing a dynamic policy management process which structures the activities required for policy management in identity and access management environments. In contrast to current approaches, it utilizes the consideration of contextual user management data and key performance indicators for policy detection and refinement and offers result visualization techniques that foster human understanding. In order to underline its applicability, this paper provides an evaluation based on real-life data from a large industrial company.

Keywords: Identity management, Policy management, Policy mining, Access control, Security management

1 Introduction

The efficient administration of employees' access to sensitive applications and data is one of the biggest security challenges for today's organizations [1]. Typically, large organizations manage millions of user access privileges across thousands of IT resources. Due to ineffective and application-specific user management, employees accumulate excessive access rights over time. As a consequence, most users are overprivileged, meaning they are assigned more permissions than necessary to perform their work. At the same time, organizational guidelines and policies can hardly be enforced in a decentralized environment. As a result, organizations implement a company-wide identity and access management (IAM) system for the centralized management of digital identities [2]. This enables organizations to implement standardized user lifecycle processes, reduce security vulnerabilities and comply with existing national and international regulations like the Sarbanes-Oxley Act [3] or Basel III [4].

In general, typical IAM systems are built on three pillars: processes, technologies and policies [5]. Core identity lifecycle processes like user (de)provisioning or access privilege management are implemented using available automation technologies. Existing products offer a variety of functionalities like identity directories for data storage, provisioning engines for user management or workflow capabilities. Both processes and technologies are controlled by a set of company-specific policies. These policies control technological aspects like data synchronization or data storage. At the same time, they are responsible for process-related aspects like access privilege management, provisioning processes, and security management within the IAM.

While available systems offer a variety of technologies and functionalities for implementing user management processes, policies have received little attention among researchers and practitioners so far. Policy management commonly still needs to be carried out manually by

*Correspondence: matthias.hummer@nexis-secure.com
[1]Nexis GmbH, Franz-Mayer-Straße 1, 93053 Regensburg, Germany
[2]University of Regensburg, Universitätsstraße 31, 93051 Regensburg, Germany

IT administrators with hardly any means for structured policy definition or ongoing policy management being available. Moreover, only static data is employed (e.g. department of an employee), letting valuable data lie fallow. As a result, only a small number of basic policies are defined and implemented in practice. These policies are commonly extracted from partly documented internal regulations and requirements and remain unchanged during system operation. This results in a situation where policies outdate over time, leading to security vulnerabilities, essentially reducing the advantages of a centralized user management. Consecutively, it is mandatory that policies evolve over time in order to reflect organizational and technological changes within a company.

In order to overcome the existing limitations, this paper introduces the dynamic policy management process (DPMP) for IAM. It provides a structured approach for policy management for IAM by applying automation technologies. On the one hand, these techniques are used in order to create a better knowledge about identity data by calculating key performance indicators (KPI) to automatically adjust policies to the current system state. On the other hand, we use them to detect new and potentially relevant policies as well as outdated policies. In contrast to existing approaches, our approach integrates the analysis of user management data as well as contextual data. The process model has been designed based on previous academic work as well as on experience gathered during our participation in several industry projects. In order to underline its applicability, we extended an existing IAM tool proposed in [6] with DPMP functionality. The tool itself provides standard IAM connectors for widely used application systems. This allowed us to facilitate available functionality and further evaluate the DPMP within a real-life use case of a large industrial company (see Section 5).

Our research methodology follows the paradigm of design science research as presented by [7] and [8]. Following the design science cycle, we derive awareness for the problem (step 1 of the design science cycle) in Section 1. In order to overcome the problems, we propose its current state of research (Section 2) together with objectives of our approach in Section 3 (2). We designed our artifact, the DPMP, in Section 4 (3). The evaluation (4) and demonstration (5) following a real-world ex-post evaluation is presented in Section 5. The adequate communication started at ARES 2015 and is further continued with this extended article in the EURASIP journal (6).

The remainder of the paper is structured as follows. In Section 2, an overview of related work is presented, and Section 3 gives a conceptual overview of current IAM systems and introduces our proposed improvement. Section 4 introduces the DPMP, while the use case based on real-world data from a large industrial company is presented in Section 5. Section 6 provides a summary and outlook for future work.

2 Related work

A large amount of research considering technological components of IAM systems and their implementation (e.g. [5, 9]), as well as their underlying access control models has been carried out [10]. However, while the importance of IAM policies in general [5] and of organizational policies in particular [11] has been acknowledged, hardly any work specifically considers the challenge of policy detection and management in large and complex environments.

In the field of policy management, researchers have proposed a variety of top-down and bottom-up policy detection approaches. Examples for discovering security policies top-down by extracting information for policy definition from existing business processes are [12–14]. Wolter et al. [12], for instance, use business process models to formulate a set of security policies using the eXtensible Access Control Markup Language. Similarly, [13] convert results from business process execution language-based processes into an role-based access control (RBAC) state [15]. Bhatti [16] specifically focus on the detection of security policies, such as separation of duty (SOD) policies. However, SOD policies only represent a small portion of the policies required in IAM systems. Bailey et al. [17] introduce a self-adaptive framework that monitors authorizations made by role- or attribute-based systems, analyzes user behavior and adapts the target systems accordingly. However, like other approaches, they focus on the detection of security policies rather than providing a guided process for comprehensive policy management in company-wide user management.

Besides the top-down approaches, several researchers have proposed bottom-up policy mining techniques [18–20]. In [20], for instance, security policies are derived from firewall and network information. Besides general policy mining approaches, the research community recently focused on mining attribute policies for attribute-based access control [21, 22] in order to ease the migration from traditional access control models such as RBAC [18, 23]. While being valuable as a technological solution, these approaches do not, among others, consider business semantics or context information required in the context of IAM to validate the correctness of suggested policies.

Additionally, these approaches focus on policy mining based on static input data. Yet, within the context of IAM, we aim to establish policy mining which uses dynamic input and thereby reduces the need for permanent policy adjustment. Due to the amount and heterogeneity of identity data, key indicators are necessary to abstract from the overall complexity and generate information about the current quality and state of the underlying IAM system.

While mining technologies are capable of finding any information within a certain set of data, this may lead to unusable output due to the improper input ("garbage in, garbage out"). By using KPIs, it is possible extract understandable and processable data out of static identity information and thus support adaptability of policies without having to change the policy itself. To our best knowledge, this part is missing in IAM policy management although it creates significant business value. Until now, IAM research mainly focus on key performance indicators for business decision support systems [24, 25]. These approaches evaluate the strategic and economic value of IAM within an enterprise and thereby compare potential benefits (e.g. reduced administration efforts or security benefits) to emerging efforts (implementation costs or operational risks). Further research aims at measuring the performance of IAM processes [26] regarding their maturity level or quality and coverage of processes.

Summing up, available bottom-up and top-down approaches mainly focus on policy detection and do not provide the structured guidance organizations requirements: (1) to implement policy discovery and recommendation mechanisms and (2) ongoing policy maintenance in IAM environments. They do not consider the integration of available context data or KPIs, decide upon the value of certain information for policy detection, or show how to transfer detected policies into daily operation. We argue that a comprehensive process model is required for structuring policy management in a company-wide IAM. Due to the complexity of IAM systems, missing support for human decision-makers reduces applicability in practical scenarios, essentially limiting the benefit of centralized user management.

3 Conceptual overview

In the following, an overview of IAM systems and their main components is provided. On this basis, we propose the extension of current IAM infrastructures using a policy mining engine for improved policy detection and recommendation. Section 4 consecutively introduces the dynamic policy management process facilitating the capabilities of the newly introduced policy mining engine throughout its structured approach for policy handling.

3.1 Identity and access management components

Typical IAM systems consist of three fundamental components (Fig. 1): IAM *data* stored in the infrastructure, tool-supported *functionalities* for executing and

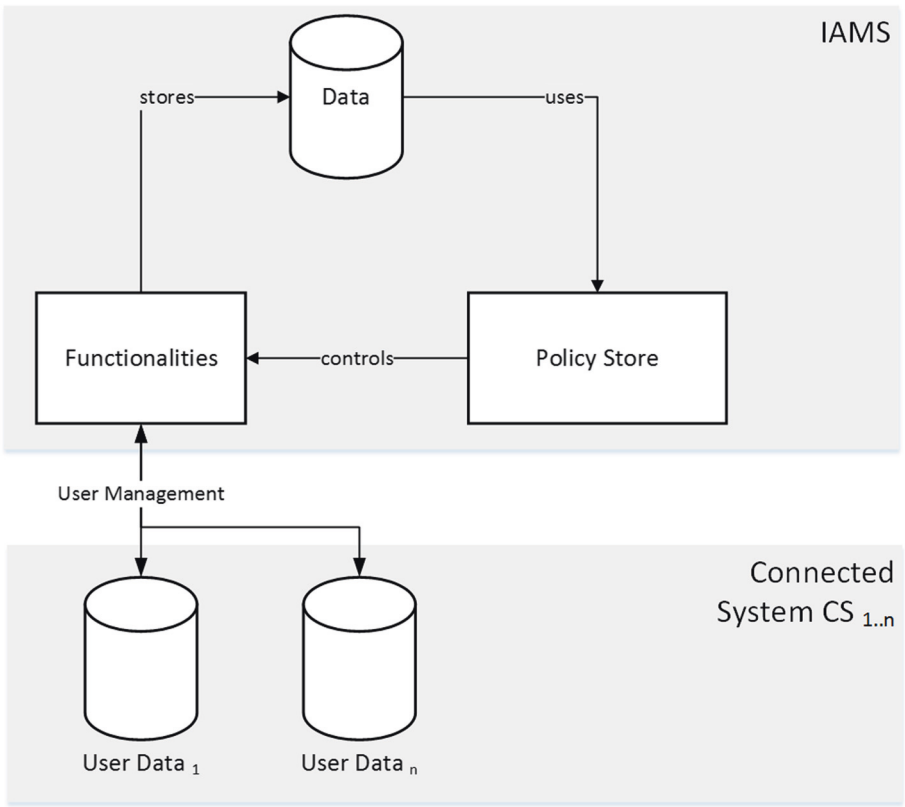

Fig. 1 Basic architecture of identity and access management systems

automating user management tasks and *policies* structuring the management of the overall IAM system itself [27].

3.1.1 IAM data

The required data within the IAM system is commonly periodically loaded from connected applications. Those can be enterprise applications having a dedicated user administration (such as the Microsoft Active Directory or SAP Enterprise-Resource-Planning (SAP ERP) systems). They, however, also can represent resources hosted by partner companies (i.e. using identity federations) or cloud-based resources. Table 1 gives a general overview of existing and used data types. Typically, one or several personnel data systems (HR system) provide employee data such as an employee's name, departmental assignment and further attributes like his or her cost center or location. At the same time, other applications provide user account information such as account identifiers and entitlement information like access privileges and related attributes (e.g. owner or description).

The IAM data coming from the various sources is linked and stored in a central database, creating new data types for a global view on identities (e.g. combining an employee's master data with his or her application-specific user accounts) and entitlements (such as business roles that group access privileges from connected applications). Both the connector technology as well as the data handling mechanisms rely on policies, e.g. for structuring the frequency of data synchronization or data correlation mechanisms.

3.1.2 Functionalities

IAM functionalities implement the logic required to operate the system and provide automated services. This includes modules for user management, access management, data handling and synchronization, or user provisioning [5, 9]. User management is concerned with managing the identity lifecycle, whereas access management provides functionality to authenticate and authorize users. Data handling and synchronization deal with integrating information from applications and exchanging data in a consistent manner. Finally, user provisioning is concerned with the allocation and revocation of user accounts and access privileges or business roles. All of these functionalities require the existence of policies guiding their mode of operation.

The last column of Table 1 underlines that current IAM systems commonly operate on the basis of information on the subject (like employee data), the object (like access privileges and applications) and the assignments between both. Thus, they are only able to process a limited static view without considering extended contextual information like an employee's activities within certain applications. In fact, most applications generate a huge amount of (audit) data such as information about a requesting entity, the affected resources, the location of access, the time and the decision of whether the request was granted or denied. Beside that, static information like assigned permissions, information about the access model or the history of an employee provides a relevant data source. Based on this source, key performance indicators may be computed e.g. for criticality or data quality which adapt the system's current state thus providing better policy input. Additionally, data such as an employee's contract status stemming from an HR system might further support policy management. We argue that considering these extended data types allows for the improved detection of access management policies.

3.1.3 Policies

Policies are used in order to define the behavior of a (software) system by using a dynamic parametrization [28]. Thus, both data and functionalities of IAM systems rely on policies for guiding their mode of operation. Among others, this has already been shown by [28] and [11], who provide an overview of various policy types and their distinct sectors of applicability. Strembeck [28] introduces three types of policies, namely authorization policies, obligation policies and delegation policies. Similarly, [11]

Table 1 Data generated within current IAMS

System	Data type	Examples	Used
HR system$_{1..n}$	Employee master data	Name, personnel number	x
	Employee context	Login state of an employee regarding different applications, vacation, criticality of entitlements	
Application$_{1..n}$	Account information	Account identifiers, account attributes (e.g. system accounts, privileged accounts)	x
	Entitlement information	Entitlement identifiers, entitlement attributes (e.g. critical entitlements)	x
	Account activity	Permission activations, activation sequences, type of permission usage, requested resources	
IAMS	Identity information	Accounts, corresponding systems, entitlements, roles	x
	Entitlement/role information	Corresponding systems, attributes	x
	Provisioning information	Requesting entities, affected resources, approving authorities, decisions	x

categorize policies into process policies, IAM policies and security policies.

The focus of authorization policies is to manage access to an object [28]. This type of policy regulates access to resources within a company and aims at increasing the security of company information and access to sensitive resources. For example, a depiction of the rule that only managers can view top-secret files falls into this category. Delegation policies are a specific set of authorization policies that allow a subject to transfer the decision-making tasks to other subjects.

Obligation policies can be divided into process policies and IAM policies. IAM policies are responsible for the design and governance of the functionality of an IAM system, whereas process policies refer to rules that describe how core business processes within organizations are executed. Examples for IAM policies are the organization's guidelines on access privilege re-certifications or provisioning policies that are used to automatically grant access to a set of resources when new employees join the company. Process policies, on the contrary, describe which permissions typically are activated together or sequentially in order to execute complete process activities.

Within the context of IAM policy management, we suggest a more application-oriented classification of policies namely explicit and implicit policies. Explicit (what can be defined as "precisely and clearly expressed or readily observable") polices are enforced by the underlying IAM system. Consequently, they cannot be bypassed by users and include a detailed definition (e.g. a script, code, rule). By default, common IAM systems already provide a broad range of implementable policies, yet these are mostly of a technical nature (like synchronization modes concerning connected applications or data storage). In order to implement more specific policies, the system itself must be customized or extended. As this is costly and requires deep technical skills, those policies are hardly changed as soon as they are implemented. Explicit policies can be categorized into security or authorization policies (actions a user is allowed to execute) which are commonly implemented in some form of access control matrix and process policies (actions which involve further interaction if the user is not directly authorized to achieve a desired result).

On the other hand, we introduce implicit (what can be defined as "implied though not directly expressed") policies which are not enforced by the IAM system itself (e.g. due to lack of suitable technical means or disproportional implementation effort). Thus, they can initially be expressed in various ways (e.g. a memo or within a dialog). Those implicit IAM policies are generally enforced by a set of stringent decisions made by operators during the lifetime of the IAM system.

Despite its importance, our experience from industry projects shows that policy management and maintenance are only rudimentary realized in practical scenarios. Policies implemented during the setup phase of an IAM system outdate over time as no technological tools or organizational guidance are available for verifying them periodically or detecting newly required policies. Defined policies are rather coarse-grained and simple. The input attributes are generally static, what can partly be attributed to the lack of available (contextual) data to identify complex polices. Another reason is the human IAM engineer's lack of understanding of how and for what applications are used by employees as well as the absence of dynamically generated data in order to allow certain policies to adapt to changes without having to change the policy itself. Additionally, scripting languages are often used to store policies. Hence, only a technically experienced personnel is able to create and refine them due to missing user interfaces.

3.2 Proposed policy management extension

In order to overcome the identified shortcomings, Fig. 2 depicts our proposed improvement. Firstly, we suggest the facilitation of currently unused contextual data for policy management. Secondly, we propose an approach to calculate policy-relevant dynamic information based on static identity data in order to improve adaptability. Thirdly, we extend policy management capabilities of IAM systems with a policy mining engine that is able to consider this contextual data during the automated detection and refinement of policies according to a structured process model (presented in Section 4).

3.2.1 Context data

According to Dey, "context is any information that can be used to characterize the situation of an entity" [29]. In today's IAM systems, almost exclusively identity and entitlement attributes are used as context data for policy decisions. Following [30], we differentiate between five types of additional context elements available in applications.

- *Activity*: Frequency and count of privilege activations as well as the amount of application data accessed.
- *Individuality*: Attributes about employees, user accounts, or access privileges data commonly available within applications (e.g. department or other attributes).
- *Relations*: Activity of similar or related employees, whereas similarity can be based on employee attributes or access privilege usage patterns.
- *Location*: The employee's location from which an activity originated. Technically, IP addresses (internal, external, VPN) are often used in this respect.
- *Time*: The date and time when a permission activation occurred, e.g. within common office hours or at night.

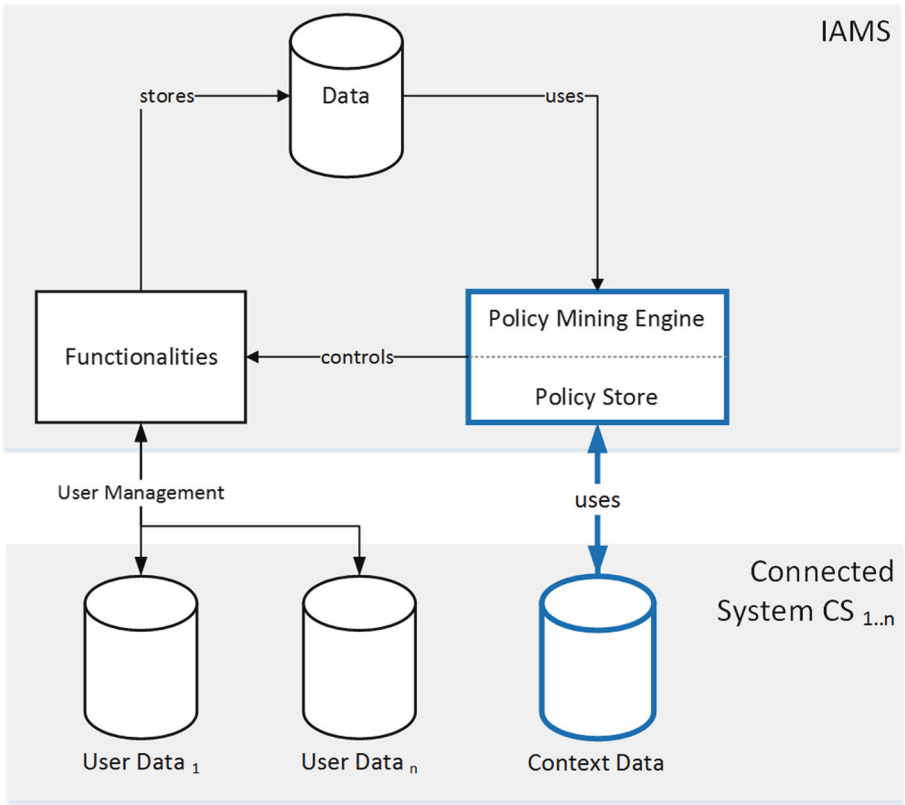

Fig. 2 Advanced architecture of identity and access management systems

3.2.2 IAM key performance indicators

The number of assignments managed by an IAM system may significantly increase over years [2]. For instance, during our evaluation (cf. Section 5.4), we analyzed an SAP ERP system with more than one million assignments of single roles to SAP user accounts, resulting in more than 36 million objects for authorization (transactions, activities, etc.). Even when such systems are carefully managed, it is hardly possible to have a detailed knowledge about every user and all of his possibilities to interact with the system based on assigned permissions. This is only one example of the growing size and complexity of modern IAM systems. While the raw data itself already is hard to comprehend due to its volume, the relations within such data are even harder to perceive. However, we argue that integrating data from the various context types explained in Section 3.2.1 can lead to a better understanding concerning the occurrence of security incidents. Due to the load of IAM data, such connections between data types need to be established in an automated way. Through detailed inspection of the integrated data, so called IAM KPIs may be defined acting as thresholds for normal behavior. Consider an example where the chief financial officer of a company is analyzing his company's net value statistics. While it is perfectly normal for him to regularly check this information, such re-occurring usage patterns integrated with the time and location of access can be good indicators for regular behavior. If such a predefined KPI reaches a value tuple that is outside of its previously common boundaries, either at runtime or ex-post, measures can be taken in order to justify abnormal behavior. Another simple KPI example are significant behavioral or entitlement changes of an employee, where there might be several reasons. Including events of the employee's history into the KPI may result in better observations about possible reasons for the changes (e.g. switched department or position, warnings) in order to generate high-quality security notifications. Thus, automatically generated information out of static data may provide an enhanced view on company events and therefore enable better automated decisions.

3.2.3 Improved policy management

To extend the policy functionality of today's IAM system, we introduce a new policy mining engine which gathers, processes and stores static and contextual data (as defined in Section 3.1.1) in order to discover KPIs and existing but not documented policies. Additionally, after monitoring and validating employees' activities for a sufficient period of time, it is able to recommend the refinement of existing

policies according to the previously defined KPIs. As an example, access patterns of employees across applications can be monitored and policies for resource access can consecutively be refined based on actual usage statistics, usage times or the criticality of access privileges.

4 Dynamic policy management process

To implement our research of an improved policy management in IAM systems in complex IAM environments, a structured process model is mandatory in order to ensure applicability. In the following section, we thus introduce the dynamic policy management process supporting organizations during their policy management activities (see Fig. 3). It consists of four phases that structure the activities required for policy management.

At first, the infrastructural setup of the policy management component within the IAM system takes place (phase 1). Input data sources are identified, and policy mining mechanisms are parametrized accordingly. Consecutively, the collection of input data is carried out (phase 2). This comprises activities like data loading, data normalization and data linking required as input data might vary regarding its currency, accuracy or provided attribute dimensions. During phase 3, the data correlation and policy mining takes place in order to differentiate between normal and outlier behavior patterns hinting at potential policy definitions and policy violations. Throughout the last step (phase 4), the results are validated and presented to human IAM engineers facilitating their organizational expertise in order to model well-designed policies.

Note that phases 2–4 of the DPMP are commonly executed in a cyclic manner while the first phase must be reentered in case the system landscape changes or other strategic changes require adjustment.

The main characteristics of the DPMP are:

- Minimizing efforts to define an initial set of policies.
- Improve the quality and adaptibility of input parameters of policies.
- Providing tool support to enable human IAM engineers to execute policy modelling and refinement.
- Integrating both actual authorization usage data and business knowledge.
- Improving IT security through continuous refinement of policies based on actual employee behavior.

4.1 Infrastructure setup

Phase 1 of the DPMP is concerned with the overall preconfiguration of the infrastructure, identifying and setting up data sources, and configuring system behavior regarding policy detection and policy recommendation.

4.1.1 Data identification and connection

Prior to the actual policy mining, available sources for contextual data need to be identified. Typical data sources are applications connected to the IAM system which store contextual data in log files. Human experts (e.g. the system administrators and IAM engineers) need to decide which contextual information from a particular application should be facilitated based on the expected business value, e.g. the potential workload reduction for user management by defining new authorization policies. For the purpose of improving the provisioning processes, for instance, the number of permission activations, the time or location (e.g. in-house, through VPN, the originating country) for each application might be of relevance.

Note that this step heavily depends on the accessibility of data and their potentially temporal availability. While data from centralized applications like SAP ERP systems might be easily accessible, contextual information collection from distributed environments (like file servers in a globally operating organization) might be cumbersome. For our approach, not all data need to be synchronized but only these within the scope of policy detection.

After the identification of available contextual information, the data connection settings need to be adjusted. The goal is an automated data synchronization based on existing connectors as well as additional application connectors (e.g. in case required contextual information stems from a system not yet connected to the IAM system). Setting up the data synchronization also includes the mapping of data from applications to the entities stored in the IAM system. Contextual data such as user account activity, for instance, needs to be related to the respective user accounts and employees.

4.1.2 Policy mining settings

After successful data selection and import, the respective data analysis configuration needs to take place. This includes the weighting of input data for automated data analysis and identification of attributes relevant for key

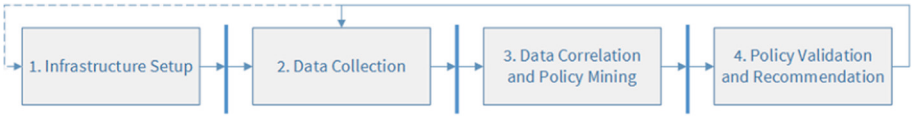

Fig. 3 Proposed policy optimization process model

indicators, as well as settings regarding the system's policy recommendation behavior. Regarding the input data weighting, human IAM engineers could e.g. decide to give more weight to data values that are constantly updated, maintained and revised and thus have a high accuracy during the consecutive algorithmic analysis.

In order to provide a better understanding and provide additional input parameters, the static access model is evaluated concerning criticalities of assignments. This is achieved using data mining techniques. Using a set of defined parameters, the calculation may be calibrated and reviewed by an human IAM engineer in order to minimize false-positives and improve confidence about the data.

Additionally, the methods of policy recommendation can be parametrized according to a given organizational scenario. Similar to approaches used for the cleansing of static access privilege assignments, for example presented in [31, 32], the DPMP requires human expert interaction after the detection of potentially reasonable policies. In case the system suggests an unreasonable large number of new policies potentially including a high rate of false-positives (detected policy suggestions which are discarded after human review), it would add an additional burden rather than create value for an organization. As a result, the system's data mining techniques need to be parametrized in order to only suggest policy definitions for selected behavior patterns.

These settings commonly require the initial analysis of input data over a reasonable period of time. Imagine the correlation of access privileges usage with employees' location data. In case the investigated privilege is only used by employees from a specific location during the period of investigation, the DPMP might recommend the definition of a provisioning policy that only assigns employees from this location to the according access privilege. If the period of investigation has been set too short, employees from other locations might also request the usage of this access privilege, essentially requiring the adaption of the defined policy.

4.2 Data collection

After successful setup of the policy management system, the data collection phase takes place. During this step, the input data is loaded, normalized and linked according to the previously defined settings. The goal is a periodic and fully automated data loading process shifting from a manual administration to an automatic machine-based execution. As a result, the latest input data are available for the automated analysis at any point in time for policy management without the need for further human interaction.

In a first step, the raw data from the relevant applications is imported and normalized. Systems which create a constant data stream require a continuous import, conversion and storage of data while other applications might only support a full data export (e.g. using the CSV file format). Furthermore, data storage types might vary among applications, requiring data normalization. Examples are an ERP application providing usage data aggregated per single day and the amount of data accessed by clients in megabytes while a file service application delivers a steady stream of data and the amount of data sent to clients in bytes. During a last data collection activity, relationships among data elements stemming from different points of time are set up. For each employee, access privilege or business role, a change history is generated. This e.g. allows for the detection of activity patterns fostering the identification of user provisioning policies.

4.3 Data correlation and policy mining

During the data correlation phase, the automated policy mining takes place. The goal is to generate recommendations for relevant policies which have not been implemented up to now. At the same time, already established policies are validated for adjustment. In this paper, it is not our goal to provide a comprehensive list of pattern detection techniques but rather aim at showing that those techniques can be applied to support policy management efforts in general. For evaluation purposes, we implemented a set of analysis techniques (see Section 5). These techniques are designed as depicted in Fig. 4.

The DPMP facilitates existing data mining technologies (e.g. clustering [33] or neural networks [34]) on the basis of existing identity information, contextual data and the various data dimensions defined during the initial setup phase (see Fig. 4). Patterns of normal and outlier behavior are automatically extracted for investigated subjects. The subject may either be a single entity or a group of entities which can be uniquely identified by a set of attributes within the context of a policy. Such an entity can be an employee, a user account within an application or a role bundling access privilege from different applications. Such data are augmented by their contextual data generated, for instance, when an entity is involved in any kind of activity.

Data mining allows for a multi-dimensional analysis facilitating sets of relevant attributes of subjects (e.g. employees, user accounts, or entitlements) and objects (e.g. amount, frequency, or criticality of data accessed). The overall goal is to identify clusters of subjects that share contextual data patterns which might in turn lead to the definition of IAM policies and the detection of outliers violating the policy.

Imagine an organization that aims at ensuring the principle of least privilege [35] in order to minimize insider misuse by overprivileged employees. Employees only are allowed to have the minimum set of access privileges required by their daily work. The DPMP in this respect

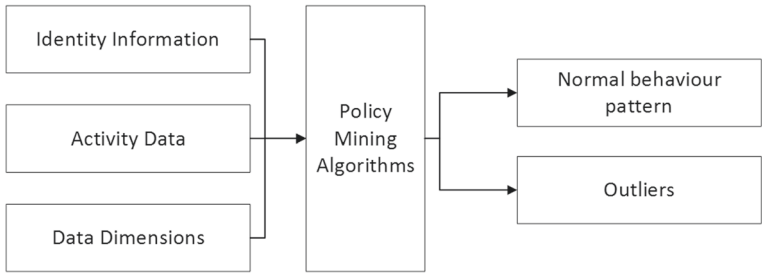

Fig. 4 Input and output of policy mining algorithms

continuously monitors existing user provisioning policies by identifying outdated access privilege assignments based on users' behavior. The example in Fig. 5 depicts the analysis of a privilege providing access to billing data within the company based on employee's location ("New York") and department ("finance"). The current provisioning policy might be refined after automated usage pattern detection identified that only employees which are assigned to the job function "clerk" actually use the respective access privileges (independent of their assigned location) while "secretaries" within the finance department in New York do not activate the access privilege at all.

Examples for the detection of anomalies (in contrast to standard usage patterns) might include entitlements for accessing financial data being activated from a VPN connection (while an according policy forbids this access) or access privileges which are used to manipulate an extraordinary amount of data.

Our approach distinguishes between the three policy types, namely security policies, process policies and IAM policies.

Every mining process is divided into three steps, namely "data construction", "data analysis" and "contextual evaluation", whereas the data analysis may go hand in hand with the contextual evaluation. During the first step, we create a suitable data structure based on availability of data and selected and weighted dimensions as well as additional information (e.g. KPIs). After that, the analysis of the data takes place. The outcome is further characterized by using available contextual data concerning involved entities. The presented approaches do not aim to present novel algorithms for the presented problems but to foster already established techniques and adjust these based on the requirements of IAM systems and their policy management components.

4.3.1 Security policy

Mining of security or authorization policies based on an available static access control matrix has been within the focus of research for some time (e.g. [18, 36]) especially since standards like ABAC [37] are heavily dependent on this construct, like this aim to extend these techniques

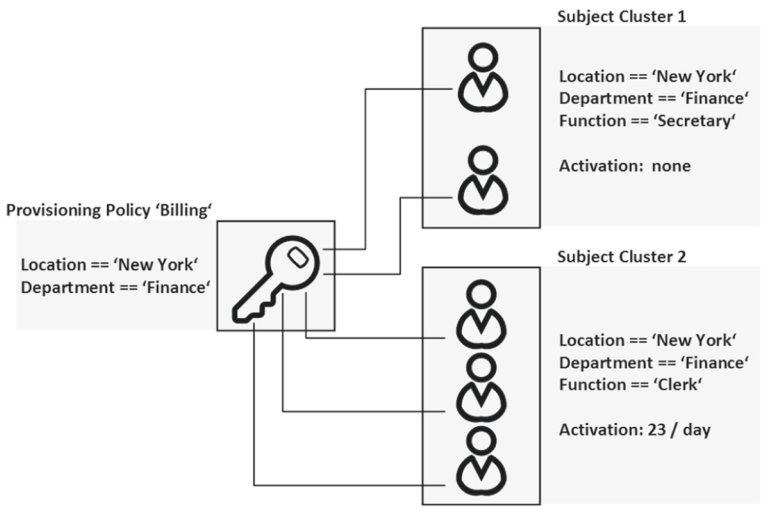

Fig. 5 Example of access privilege activation analysis

by enhancing the input data and the characterizations of output data using contextual information.

The first step is to analyze the existing access control matrix based on semantic analysis techniques and usage statistics. Semantic analysis [31, 32] introduce a classification for every assignment based on the examination in context of other entities within the given scope. A simple example would be if only one employee within the "development" department owns the entitlement "access marketing share". As a result, this permission assignment might probably be invalid and should be revoked. We use these techniques to sort out potentially invalid assignments which create noise during policy mining. Similarly, we identify and exclude unused entitlements (with an adjustable period of time) and recommend manual review by human IAM engineers.

For the authorization policy mining, we facilitate available algorithms (like those proposed at [16, 18, 21]). These algorithms operate on different types of data, e.g. log files, roles or user permission assignments, and can be adjusted according to the available access control model and available data sources.

During the last step, we analyze every mined policy according to its usage profiles. This includes attributes like location, time, consumed CPU resources or the amount of data read or written. These profiles are analyzed using classification techniques (e.g. [33]) in order to reveal normal and abnormal utilization behavior. Consider the example of an entitlement to modify data within an application. The amount of data typically modified during normal manual operation can be classified e.g. financial data are usually not modified throughout activities creating gigabytes of traffic. This enables human IAM engineers to evaluate usage profiles of entitlements or authorization policies according to compliant operation of applications.

4.3.2 Process policy

Process policies represent a subset of obligation policies. They define constraints for actions within a process which need be carried out in order to achieve a desired business value of which a user is not directly authorized to. Within context of IAM, for example, this could be the obtaining of an approval to assign a specific access right or a re-certification. In order to identify process policies, we aim to identify events which trigger such processes as well as necessary checkpoints or nodes (e.g. the head of department's approval) which need to be passed in order to achieve a desired result.

For the transformation of activities into structured processes, we use business process mining (BPM) techniques (as proposed in [38]). For data construction, we use the concept of trace clustering which divides activity logs into traceable clusters or in other words single process iterations. In the context of IAM, this could be a ticket number or a process id in relation with a specific request type. This information are subsequently mapped into a process representation (e.g. BPMN) for further analysis. Information about processes could theoretically be extracted directly from a business process management system. Yet, the goal is to create a detailed overview about the status quo within the IAM system (independent of how the system should actually be used) and possibly create a comparison to already defined processes.

During the next step, we try to derive a detailed characterization of every process node including decision-making entities as well as the respective context of the decision. We aim to identify similarities between the decisions (e.g. every re-certification was done during business hours from within the IT building by an employee within the same department and attribute "head of department") as well as outliers (e.g. every approval of this entitlement was done by an employee with the attribute "entitlement owner" during business hours, while one approval was done at 22:00 pm by an admin account). In order to achieve this, we firstly have to generalize information as far as available. For example regarding the time of actions, we may distinct between business hours and closing time, location between off- and onsite, devices could be separated into business owned devices, private devices and hybrid usage. Using this compression, we are able to reduce the number of possible definitions for each process node. After that, we classify affected entities per process step using different, individually weighted attribute permutations as input.

During step three, we create an extended process definition. For every process step, the created classifications are analyzed. If a classification which includes every entity who finished the respective step is available, its attribute combination is used as a possible definition for the step. If no suitable cluster can be identified, all clusters are used as possible definitions and thereby an extended manual review is necessary.

4.3.3 Implicit IAM policies

As mentioned above, IAM policies are responsible for the design and compliant operation of IAM systems. Yet by far not every IAM policy is technically implemented. The IT of today's companies is forced to adapt business changes and support the business in an optimal manner. Thus, directly customizing the IAM system according to every business change is hardly executed in practice because of the resulting efforts. Due to these reasons, we do not aim to exhaustively mine all possible IAM polices which are currently in place and technically enforce them as it would impose rigid restrictions to the system and potentially have a negative impact on business processes. However, we propose a mechanism which allows the system to learn current IAM policies and create recommendations

regarding system operation. Consider the example of an international help desk who is in charge of processing orders regarding the IAM system (e.g. the request for assignment of an entitlement or access to a specific application) with all requests requiring manual review. By generating recommendations out of previous decisions, approvers can be supported during the review process.

Our first step is to classify users of the system according to their weighted attributes (e.g. [33]). This enables us to derive an overview of which types of users are managed by the system. If applicable, there may be several different classifications if the company is structured in a complex way. After that, we derive a set of actions (e.g. requests, authentications) carried out by those types of users. These partitioned user sets combined with affiliated contextual data and the history of activities allows us to use a context-aware recommender system (e.g. [39, 40]). Standard recommender systems in general aim at providing suggestions for items which are considered to be useful for a user [40], whereas context-aware recommender systems operate on tuples in the form of $< user, item, context, rating >$. In the context of IAM, users are the company's employees and items represent achievable resources (e.g. entitlements or files). As soon as the calculated rating extends the defined threshold, a positive recommendation is given, otherwise a negative one. As context-based frameworks for recommender systems take attributes like activity, location, user information and related resources into consideration [40], they consequently meet the requirements of an IAM policy recommendation system.

4.4 Policy validation and recommendation

After successful data correlation and policy mining, a set of potentially relevant policies (e.g. provisioning policies changing the current access control state) has been identified. As IAM systems and connected applications manage a huge amount of data, a high number of potentially relevant policies might be detected by each DPMP iteration. These policy candidates need to be validated by the policy management system before being communicated in an appropriate manner to human IAM engineers for refinement. Policy validation thus can observe the underlying rule for every detected policy over a certain period in time before it is recommended to a human IAM engineer. In case a policy suggestion is based on usage activity patterns, for instance, these patterns can be validated over a period of 1 month. In case the pattern changes during the investigation period, the policy suggestion itself can be revoked.

Policy mining is limited to generating a set of policy suggestions based on classifications of subjects together with their behavior based on contextual data and history. As a result, the focus during the last phase of the DPMP lies on the presentation of results in an intuitive and human-understandable way in order to enable the IAM engineer to easily derive appropriate actions. Visualizations can be based on techniques like charts or data tables. From our practical experience, it is essential to include the visualization of the reasons why a certain policy suggestion has been created. In case ambiguous or mutually exclusive rules have been identified, this information has to be included in the result presentation as well. A human IAM engineer might, for instance, be informed that accepting one policy suggestion might lead to the violation of another already implemented policy. He then might be able to decide whether the old policy is outdated while the new policy suggestion should be activated.

Again, it is not the goal of this paper to provide a comprehensive list of potential visualizations or rule definition scenarios but rather underline the importance of a dedicated result refinement phase including human interaction as a cornerstone of ongoing policy management in IAM.

In this section, we proposed the dynamic policy management process which enables organizations to gather a deeper understanding of its IAM, the (contextual) data and the quality of currently implemented policies as well as potential policy suggestions. Based on company-specific settings, the DPMP is able to import the necessary input data, identify patterns of standard subject behavior and support human IAM engineers during policy definition and refinement.

5 Evaluation

In this section, we evaluate the applicability of the DPMP in a real-world scenario. The evaluation is based on data stemming from the SAP ERP system and the IAM system of a globally operating manufacturing company with more than 12,000 internal and 4000 external employees. A total number of 8021 active user accounts, 3925 single roles, 762 composite roles and 1,180,962 access privilege assignments from the SAP ERP system were initially imported and anonymized. For the following evaluation, the period under observation comprised 5 weeks during which daily re-imports took place.

Increasing audit requirements force the company to improve IAM policy management. Up to now, only rudimentary provisioning and access re-certification policies have been defined due to missing tool support and knowledge about the underlying data. As a result, a policy detection project has been initiated. Its main goals are:

1. The consideration of contextual data and KPI definition from the SAP ERP system for policy generation
2. The semi-automated detection of new and potentially relevant provisioning and re-certification

policies as well as the identification of loosely defined and hence insecure existing policies
3. Providing appropriate visualizations of detected policies to support human IAM engineers

While requirement (1) corresponds to phase 1 and 2 of the DPMP, (2) relates to its data correlation and policy mining phase (phase 3). Requirement (3) deals with the presentation of discovered policies according to phase 4 of the DPMP. Even though we executed numerous policy detection activities, we focus on two specific examples for evaluation purposes in the remainder. Firstly, the analysis of access privilege activations has been compared to the static distribution generated by the current provisioning policy in the IAM system (corresponding to phase 2 of the DPMP, see Section 5.3). Secondly, detected access privilege activation frequencies were visualized in relation to the amount of data objects modified (i.e. data within the SAP ERP system) for investigation by a human IAM engineer (corresponding to phases 2 and 3 of the DPMP, see Sections 5.3 and 5.4).

Note that a comparative evaluation of our prototype-based approach with manually executing policy detection and recommendation cannot be executed. This is due to the inapplicability of a manual examination of the several hundred thousands of access privilege assignments and the available large amount of contextual information.

5.1 Infrastructure setup

To address requirement (1), at first, context data available in the SAP ERP system was analyzed (step 1 of phase 1). Using the classification technique for context data from Section 3.2.1, the following information on user behavior was extracted and mapped: number and frequency of read and write permission usage and amount of transferred data (activity) for each account (individuality) per day (time) and the corresponding IP address (location). Subsequently, policy mining parametrization was conducted (step 2 of phase 1). Initially, the set of prototypical implemented algorithms (including data classification mechanisms and statistical distribution analysis) were applied using a default configuration. On this basis, distinctive properties of the imported data set became apparent. For instance, due to SAP ERP system limitations, user behavior can only be extracted on a daily basis. Thus, algorithms need to be configured to identify permission usage irregularities per day (e.g. suspicious permission activations on weekends) but not within the course of a single day (such as off-time activities). Furthermore, data types were weighted in cooperation with a human system expert, emphasizing the importance of data types such as IP address and employee status information during the following analyis.

5.2 Data collection

After successful configuration and parametrization, the data collection took place (phase 2 of the DPMP). We implemented a software wizard to ease the import of raw data types onto the internal data structures of the extended IAM tool (see Fig. 6) in an automated manner. The wizard shows an excerpt of available contextual data which can be extracted from an SAP ERP system. Available contextual data from applications strongly vary (e.g. contextual data for an active directory may be derived from connected share systems). The wizard shows an

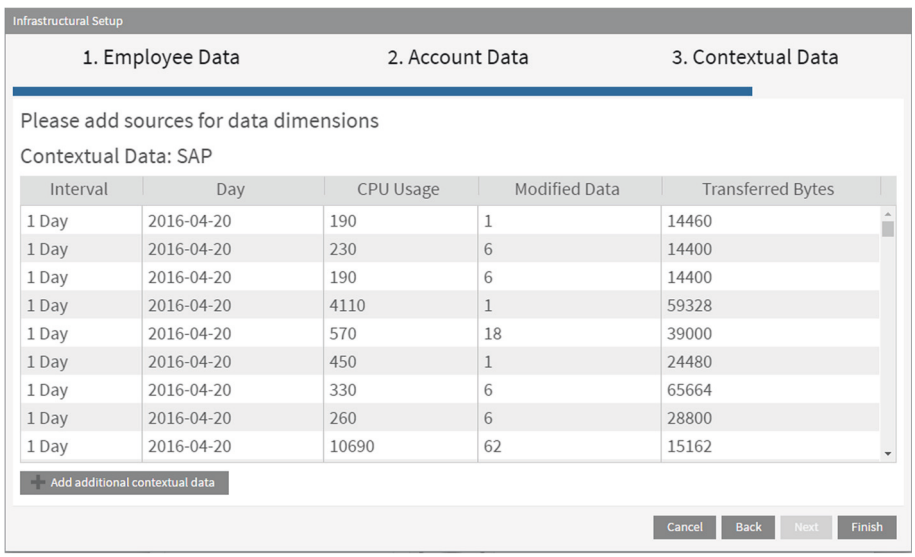

Fig. 6 Infrastructure setup wizard

excerpt of the broad variety of contextual data which can be extracted from applications and used for the policy mining step. Additionally, data from the SAP ERP system was mapped onto existing user management data from the IAM system. SAP user account activities, for instance, were related to the respective employees' identities. As a result, a total number of 6,214,422 records from 36 days containing contextual information as well as user management data from the SAP ERP system were gathered and mapped using our daily data import functionality.

5.3 Data correlation and policy mining

During the third phase of DPMP the actual policy mining was conducted in order to address project requirement (2). Using our implemented policy mining algorithms, we were able to detect standard usage patterns potentially leading to the definition of new policies as well as the refinement of currently implemented policies.

Concerning the first exemplary case, we computed the distribution of static assignments of access privileges among the top level departments of the company and compared these to their actual activation information. Table 2 shows the distribution of access privilege P_1 across top-level departments of the company and its actual activation in these departments. As can be seen, nearly half of the employees that are assigned to P_1 are working in the department D_3. The access privilege is almost exclusively used (99.96 %) in this department, while only a small number of activations (0.04 %) stems from department D_1. This indicates that access privilege P_1 might only be required for tasks conducted in department D_3. Thus, a refinement of the existing provisioning policy that additionally requires employees to work in department D_3 in order to obtain this access privilege is recommended. This restructuring might lead to a reduction of the number of overprivileged employees, thereby strengthening IT security.

In summary, out of the company's total 3925 single roles defined in the SAP ERP system, we identified 382 (i.e. 9.7 %) which—though being assigned to employees in a particular department—were hardly activated (activation frequency for the respective department is below 1 %). In an ongoing effort, these results are discussed with the company's IAM engineer in order to improve existing provisioning policies and refine existing SAP role definitions leading to access privilege revocation.

Table 2 Static distribution and actual use of access privilege P_1

Department	Distribution of static assignment (%)	Activation frequency (%)
D_1	21.15	*0.04*
D_2	24.39	0.00
D_3	45.08	*99.96*
D_4	4.82	0.00
D_5	0.72	0.00
D_6	1.54	0.00
D_7	2.3	0.00

For our definition of KPIs, we intensively examined a criticality value for every employee based on his currently assigned entitlements. An employee was defined as critical as soon as he owned permissions which are not common for his position within the enterprise. The more uncommon an entitlement is (e.g. because he owns multiple times as many permissions as other employees within his departments), the higher the criticality value. By calculating such value for each employee, the policies that are addressing privileged employees can be redefined. In discussions with the company, we agreed on primarily taking the employee's contextual data of permissions into account (in this case his department and other employees provisioned with similar access rights). For this effort, we used a set of parametrized anomaly detection algorithms for outlier detection algorithms [41] for the criticality determination of user permission assignments. Our applied algorithms measure the distribution of permission assignments among a defined set of users. We use different sets of input parameters ranging from a high to a low detection rate. The lower the detection rate, the higher the criticality value of the assignment. A simple example for such an algorithm would be to detect all entitlements assigned to not more than a certain percentage of employees. For each of our algorithms, we created a set of parameters varying from a severe to a slack detection rate. According to the quantity of re-occurrences of results throughout different parameter sets, we categorized the assignments of the employees on a Likert scale. This ranges from *uncritical* to *very critical*, thus providing an easy to understand overview about the current access model. Consider the following simplified example demonstrating the functionality: an employee from the finance department is switching positions within the company and is now working for the marketing department. Due to lack of correct revocation policies, the employee retains some of his old permissions. In our approach, each of these permission assignments would be detected by each run of the algorithms with all of the previously calibrated parameter sets. As even the strictest parameter set (i.e. the set that tolerates least errors in the data) together with all others flags this assignment as critical, its overall criticality is set to *very critical*. The resulting distribution for the assessment of all of the company's assignments is depicted in Table 3.

Fostering these results enables the system to automatically classify each employee concerning his criticality. For the employee categorization, we followed the maximum principle according to the BSI Grundschutz [42] which

Table 3 Criticality of assignments based on static analysis

Criticality	Number of assignments	Percentage (%)
Uncritical	1,087,939	92.12
Very low criticality	34,494	2.92
Low criticality	31,888	2.70
Critical	13,792	1.17
Very critical	12,851	1.09

states that the security level of an object should be as high as the highest of its associated resources (in our case the most critical user permission assignment):

$$\text{Criticality}_{\text{employee}} = \text{Max}\left(\text{criticality}_{\text{UPA}}\right).$$

The results of our criticality calculations are as follows. The data depict a very even distribution of assignments among the overall company having only 1.08 % assignments with a *high criticality* level. Nevertheless, there are 12,851 assignments affected, which may impose severe security risks upon the enterprise (according to our calculation that we established in conformance with the company). Concerned were a total number of 428 highly critical user accounts. These data can now be used in order to enhance affected policies for user management e.g. by employing more frequent re-certifications ([43]) or four-eye prinicples.

In order to address project requirement (3), we aimed to derive usage patterns of permissions based on contextual data. As mentioned above, the system currently manages over one million permission assignments (an average of about 147 assignments per user) with more than 30 million assignments possible. Therefore, we aim to qualify permission assignments based on activities carried out by users. Single roles usually conform with a well-defined set of actions which they enable a user to execute. Consequently, these actions generate a similar fingerprint concerning usage profiles (e.g. data modified or read) within a predefined period of time. We aim to identify these fingerprints by applying our security policy mechanisms. The first step is to set up a suitable data set for the analytics which consists of all user permission assignments (UPA) and a set of contextual data concerning permission activations aggregated per day (see Fig. 6). We combined these data to a set of vectors in the form of:

$$V_{\text{activity}} = \{\text{UPA}, \text{datamodified}, \text{dataread}\}.$$

For these data, we created clusters using [44]. Accordingly, we are able to classify the usage of every permission assignment on a daily level. As described above, we computed a criticality value for all user permission assignments. This enables us to combine classified usage profiles (as depicted in 7) with the criticality value of each assignment. As these results are not directly usable by human IAM engineers, suitable visualization techniques have to be applied.

5.4 Policy validation and recommendation

In order to further address project requirement (3), previously detected standard usage patterns need to be validated and visualized for human refinement. Figure 7 depicts a screenshot from our extended IAM tool which uses a bubble chart visualization in order to display detected usage patterns. The x-axis corresponds to the amount of data that has been "modified" by an employee's access privilege activations on a single day, while the y-axis denotes the amount of data being "read". During phase 1 of the DPMP, thresholds were defined for highlighting power users, i.e. employees which either read or modify large amounts of data within the SAP ERP system. In the given example, an employee has been marked as a power user (orange colored highlighting) if he either read more than 10,000 MB or modified more than 200,000 data sets

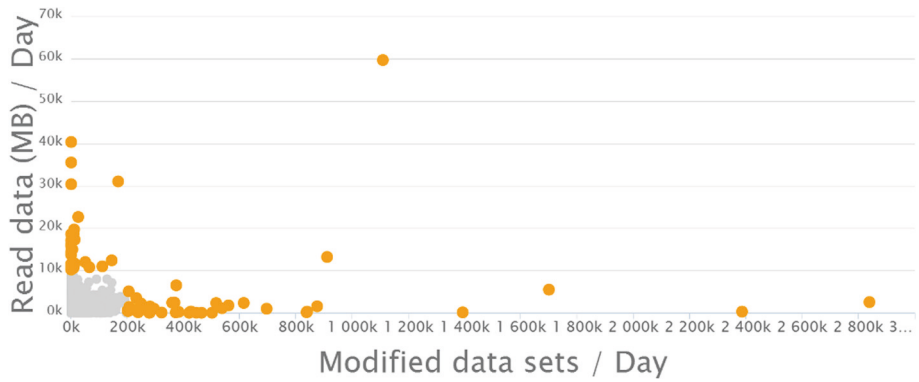

Fig. 7 Detection of SAP power users (http://www.nexis-secure.com)

per day. Bubbles in the lower left area of Fig. 7 correspond to average system users, while highlighted bubbles in the other areas correspond to power users. In the given example, 63 power users were identified for the interval of our investigation. Out of these 63 power users, we identified three users who activated assignments which have been marked critical or very critical during our KPI analysis. A human IAM engineer could use this information for defining a new re-certification policy that demands a periodic assessment of all power users' access privileges. In contrast to standard SAP users whose access privileged are re-certified once a year, power users might be re-certified more frequently in order to reflect their criticality value.

In summary, the evaluation based on data from an SAP ERP system presented in this section of the paper underlined the applicability of the DPMP for structured policy management in practice. Based on the prototypical extension of an existing IAM tool, we were able to import previously unused contextual data, identify clusters of standard as well as outlier usage behavior and visualize the gathered results. Within the company, the results increased management attention by providing in-depth insight into the current access control state and its guiding policies. At our partner's side, efforts for evaluating the application of the DPMP in a periodic manner (daily operation), the extended analysis of further applications, and the adaption of existing IAM policies are currently made.

6 Conclusions

Over the last decades, company-wide IAM systems have become a key element for controlling users' access to resources in medium to large-sized enterprises. They offer means for a centralized enforcement of standardized user management processes and policies. Despite their importance, the management of IAM policies commonly still needs to be executed manually. While current research concentrates on mechanisms for policy detection and enforcement, the complexity of user management in large environments rather requires a structured and applicable process for policy management. Human IAM engineers need to be supported with guidance and automation during the detection, implementation and refinement of IAM policies.

In order to improve the current situation, we presented the dynamic policy management process which structures the activities during policy management into four phases. It facilitates a mining engine which generates policy recommendations based on contextual data of employees and further presents gathered results to human IAM engineers. In order to underline the practical relevance and applicability of our contribution, we conducted a practical case study within a large industrial company and its ERP system managing several thousands of users and more than one million access privileges.

We showed and also experienced from our industrial projects that policy management is an important task within modern IAM architectures as it provides an anchor for the system to work properly and secure in a time where enterprises begin to realize that the traditional castle approach of their IT imposes several risks e.g. due to cloud computing, work anywhere, IoT or Industry 4.0. Due to these developments, IAM will become even more important and therefore need a stable basis to work on.

For future work, we plan to extend the DPMP in order to improve the representation and management of policy recommendations. Practical experience shows that a high amount of potentially conflicting recommendations increases manual efforts of human role engineers and requires an in-depth understanding of the underlying data. In the future, we hence aim at providing an analysis of policy interdependencies in order to overcome this limitation. We additionally aim at extending our prototype implementation and evaluate the DPMP throughout further practical use cases considering contextual data from decentralized applications.

Acknowledgements
The research leading to these results was partly supported by the "Bavarian State Ministry of Education, Science and the Arts" as part of the FORSEC research association.

Authors' contributions
Firstly, we suggest the facilitation of currently unused contextual data for policy management. Secondly, we propose an approach to calculate policy-relevant dynamic information out of static identity data in order to improve adaptibility. Thirdly, we extend policy management capabilities of IAMS with a policy mining engine that is able to consider this contextual data during the automated detection and refinement of policies according to a structured process model. Fourth we demonstrate the feasibility of our approach based on real-life data. The design was carried out by MH, MN and MK. MK fit the article into related work, MN worked on the conceptual overview. MH and MK established the DPMP along with its algorithmic approach. MH and LF carried out the evaluation. GP induced the research which lead to this article. All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Received: 11 February 2016 Accepted: 2 August 2016
Published online: 15 August 2016

References

1. A Hovav, R Berger, Tutorial: identity management systems and secured access control. Commun. Assoc. Inf. Syst. **25**(1), 42 (2009)
2. A Cleven, R Winter, in *Enterprise Business-Process and Information Systems Modeling. Lecture Notes in Business Information Processing*, ed. by T Halpin, J Krogstie, S Nurcan, E Proper, R Schmidt, P Soffer, and R Ukor. Regulatory Compliance in Information Systems Research – Literature Analysis and Research Agenda, vol. 29 (Springer, Berlin Heidelberg, 2009), pp. 174–186
3. United States Code, Sarbanes-Oxley Act of 2002, PL 107-204, 116 Stat 745 (2002). https://www.sec.gov/about/laws/soa2002.pdf. Accessed 11 Aug 2016
4. Basel Committee on Banking Supervision, Basel III - A global regulatory framework for more resilient banks and banking systems (2011). https://www.bis.org/publ/bcbs189.pdf. Accessed 11 Aug 2016

5. L Fuchs, G Pernul, in *The Second International Conference on Availability, Reliability and Security, 2007: ARES 2007*. Supporting compliant and secure user handling—a structured approach for in-house identity management (IEEE Computer Society, Los Alamitos, 2007), pp. 374–384
6. L Fuchs, M Kunz, G Pernul, in *European Conference on Information Systems (ECIS)*. Role model optimization for secure role-based identity management, (2014)
7. K Peffers, T Tuunanen, CE Gengler, M Rossi, W Hui, V Virtanen, J Bragge, in *Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006)*. The design science research process: a model for producing and presenting information systems research (M. E. Sharpe, Inc., Armonk, 2006), pp. 83–106
8. AR Hevner, ST March, J Park, S Ram, Design science in information systems research. MIS Q. **28**(1), 75–105 (2004)
9. D Royer, in *Proceedings of the IFIP/FIDIS summer school on "The future of identity in the information society*. Enterprise identity management—what's in it for organisations (Springer, Berlin Heidelberg, 2008), pp. 403–416
10. L Fuchs, G Pernul, R Sandhu, Roles in information security—a survey and classification of the research area. Comput. Secur. **30**(8), 748–769 (2011)
11. L Fuchs, G Pernul, Minimizing insider misuse through secure identity management. Secur. Commun. Netw. **5**(8), 847–862 (2012)
12. C Wolter, A Schaad, C Meinel, in *Web Information Systems Engineering–WISE 2007 Workshops*. Deriving XACML policies from business process models (Springer, Berlin Heidelberg, 2007), pp. 142–153
13. J Mendling, M Strembeck, G Stermsek, G Neumann, in *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2004. WET ICE 2004. 13th IEEE International Workshops on*. An approach to extract rbac models from BPel4Ws processes (IEEE Computer Society, Los Alamitos, 2004), pp. 81–86
14. A Baumgrass, S Schefer-Wenzl, M Strembeck, in *IEEE*. Deriving process-related RBAC models from process execution histories, (2012), pp. 421–426
15. RS Sandhu, EJ Coyne, HL Feinstein, CE Youman, Role-based access control models. IEEE Commun. **29**(2), 38–47 (1996). doi:10.1109/2.485845
16. R Bhatti, E Bertino, A Ghafoor, X-federate: a policy engineering framework for federated access management. IEEE Trans. Softw. Eng. **32**(5), 330–346 (2006). doi:10.1109/TSE.2006.49
17. C Bailey, DW Chadwick, R de Lemos, in *IEEE 9th International Symposium on Dependable, Autonomic and Secure Computing*. Self-adaptive authorization framework for policy based RBAC/ABAC Models (IEEE Computer Society, Los Alamitos, 2011), pp. 37–44
18. Z Xu, SD Stoller, in *Data and Applications Security and Privacy XXVIII*. Mining Attribute-Based Access Control Policies from Logs (Springer, Berlin Heidelberg, 2014), pp. 276–291
19. A Baumgrass, in *Availability, Reliability and Security (ARES), 2011 Sixth International Conference On*. Deriving current state RBAC models from event logs (IEEE Computer Society, Los Alamitos, 2011), pp. 667–672
20. H Safaa, C Frédéric, C-B Nora, A Vijay, M Stéphane, in *9th International Conference on Information Systems Security*, ed. by A Bagchi, I Ray. Policy Mining: a Bottom-Up Approach Toward a Model Based Firewall Management (Springer, Berlin Heidelberg, 2013), pp. 133–147
21. J Lopez, R Oppliger, G Pernul, Authentication and authorization infrastructures (AAIs): a comparative survey. Comput. Secur. **23**(7), 578–590 (2004)
22. VC Hu, D Ferraiolo, R Kuhn, A Schnitzer, K Sandlin, R Miller, K Scarfone, Guide to attribute based access control (ABAC) definition and considerations. NIST Spec. Publ. **800**, 162 (2014)
23. Z Xu, SD Stoller, in *Emerging Technologies for a Smarter World (CEWIT), 2013 10th International Conference and Expo on*. Mining attribute-based access control policies from RBAC policies (IEEE Computer Society, Piscataway, 2013), pp. 1–6
24. D-W-ID Royer, M Meints, Enterprise identity management—towards a decision support framework based on the balanced scorecard approach. Bus. Inf. Syst. Eng. **1**(3), 245–253 (2009)
25. MC Mont, Y Beresnevichiene, D Pym, S Shiu, in *Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP*. Economics of identity and access management: providing decision support for investments (IEEE Computer Society, Piscataway, 2010), pp. 134–141
26. D Royer, M Meints, Planung und Bewertung von, Enterprise identity managementsystemen. Datenschutz und Datensicherheit-DuD. **32**(3), 189–193 (2008)
27. J Pato, OC Center, *Identity Management: Setting Context*. (Hewlett-Packard, Cambridge, 2003)
28. M Strembeck, *Engineering of Dynamic Policy-Based Systems: A Policy Engineering of Dynamic Policy-Based Systems: Language Based Approach*. (Habilitation Thesis, WU-Wien, 2008)
29. AK Dey, Understanding and using context. Pers. Ubiquit. Comput. **5**(1), 4–7 (2001)
30. A Zimmermann, A Lorenz, R Oppermann, in *Proceedings of the 6th International and Interdisciplinary Conference on Modeling and Using Context, CONTEXT'07*. An Operational Definition of Context (Springer, Berlin Heidelberg, 2007), pp. 558–571
31. L Fuchs, C Broser, G Pernul, in *Availability, Reliability and Security, 2009. ARES'09. International Conference On*. Different approaches to in-house identity management—justification of an assumption (IEEE Computer Society, Piscataway, 2009), pp. 122–129
32. A Colantonio, R Di Pietro, A Ocello, NV Verde, A new role mining framework to elicit business roles and to mitigate enterprise risk. Decis. Support. Syst. **50**(4), 715–731 (2011)
33. J MacQueen, *et al*, in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. Some methods for classification and analysis of multivariate observations, vol. 1, (Oakland, 1967), pp. 281–297
34. T Kohonen, An introduction to neural computing. Neural Netw. **1**(1), 3–16 (1988)
35. JH Saltzer, MD Schroeder, The protection of information in computer systems. Proc. IEEE. **63**(9), 1278–1308 (1975)
36. Z Xu, SD Stoller, Mining attribute-based access control policies. IEEE Trans. Dependable Secure Comput. **12**(5), 533–545 (2015)
37. T Priebe, W Dobmeier, B Muschall, G Pernul, in *Sicherheit 2005: Sicherheit - Schutz und Zuverlässigkeit, Beiträge der 2. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft Für Informatik e.V. (GI), 5.-8. April 2005 in Regensburg*. ABAC - Ein Referenzmodell für attributbasierte Zugriffskontrolle (GI, Bonn, 2005), pp. 285–296
38. L García-Bañuelos, M Dumas, M La Rosa, J De Weerdt, CC Ekanayake, Controlled automated discovery of collections of business process models. Inf. Syst. **46**, 85–101 (2014)
39. K Verbert, N Manouselis, X Ochoa, M Wolpers, H Drachsler, I Bosnic, E Duval, Context-aware recommender systems for learning: a survey and future challenges. IEEE Trans. Learn. Technol. **5**(4), 318–335 (2012)
40. F Ricci, L Rokach, B Shapira, in *Recommender Systems Handbook*. Introduction to recommender systems handbook (Springer, Berlin Heidelberg, 2011), pp. 1–35
41. G Pernul, L Fuchs, Reducing the risk of insider misuse by revising identity management and user account data. J. Wirel. Mob. Netw. Ubiquit. Comput. Dependable Appl (JoWUA). **1**, 14–28 (2010)
42. B für Sicherheit in der Informationstechnik, BSI-Grundschutz Katalog (1996). https://www.bsi.bund.de/EN/Topics/ITGrundschutz/itgrundschutz_node.html. Accessed 11 Aug 2016
43. C Richthammer, M Kunz, J Sänger, M Hummer, G Pernul, *Dynamic Trust-based Recertifications in Identity and Access Management*. (IEEE Computer Society, Piscataway, 2015)
44. JA Hartigan, MA Wong, Algorithm AS 136: A k-means clustering algorithm. J. R. Stat. Soc. Ser. C (Appl. Stat.) **28**(1), 100–108 (1979)